Beyond Source Code

Mike Amundsen,
API Academy / CA
@mamund
Introduction
Your Guide to API Design & Implementation Best Practices

API Academy delivers free online lessons and in-person consulting services covering essential API techniques and tools for business managers, interface designers and enterprise architects.

- What is an API?
  Get an overview of what an API is and what it does, to help you realize the business value of APIs.

- API Design Basics
  Understand the API architecture process and learn basic design and implementation best practices.

- Web API Architectural Styles
  Get a detailed overview of the main architectural styles for Web and mobile API design.

- Choosing a Solution
  Choose between the various solutions that offer the basic components for enterprise API Management.
Beyond Source Code

• Computing
• Communicating
• Scaling
• Artisinal Luddites
• The Future of Code
Computing
“Figure out how the machine works and then figure out how to program it.”

- Kay McNulty, ENIAC Team
Mental Models affect the way we see and interpret reality. They are like the filters through which we see the world.
There was no source code...
There was no source code…
The program was the **machine**.
TYPICAL COBOL.

000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. PAYROLL.
000300 AUTHOR. JOHN DOE.
000400 DATE. APRIL 5TH 1960.
000500 REMARKS.
000600 INPUT FROM RUN 1 AND OUTPUT TO RUN 25.
   THIS PROGRAM PROCESSES SALARIED EMPLOYEES ONLY.

0002000 ENVIRONMENT DIVISION.
0002100 CONFIGURATION SECTION.
0002200 SOURCE-COMPUTER. COMPUTER NAME.
0002300 OBJECT-COMPUTER. COMPUTER NAME.
0002400 SPECIAL NAMES. HARDWARE NAME.

0003000 INPUT-OUTPUT SECTION.
0003100 FILE CONTROL. SELECT FILE-NAME 1
0003200 SELECT FILE-NAME 2 SELECT 
0003300 1-O CONTROL. APPLY 

0004000 DATA DIVISION.
0004100 RD MASTER-PAYROLL, LABEL RECORDS ARE
0004200 STANDARD. DATA RECORDS ARE MASTER-
0004300 PAY. SEQUENCED ON BADGE-NUMBER.
0004400 OF MASTER-PAY SIZE IS 180 CHAR-
0004500 ACTERS. CLASS IS ALPHAMERIC.
0004600 02 BADGE-NUMBER SIZE IS 12
0004700 CHARACTERS. PICTURE IS
0004800 AAAAXX9999999.

0006000 PROCEDURE DIVISION.
0006100 COMPUTATIONS SECTION.
0006200 UPDATE-MASTER. MOVE ADJUSTED-PAY TO
0006300 NET-PAY. ADD GROSS-PAY TO GROSS-
0006400 YEAR-TO-DATE. WRITE UPDATED-
0006500 MASTER-PAY. READ MASTER-PAYROLL
0006600 RECORD.

******
Now there is source code…
The machine is the \textit{code}.
Communicating
Melvin Conway
Project-Based Organizations
“Project-based organizations revolve around the concept that a group of individuals or firms join together with the explicit purpose of producing a tangible set of outputs”

-- Paul Chinowsky, EPOJ 2011
“How Do Committees Invent?”
HOW DO COMMITTEES INVENT?

by MELVIN S. CONWAY

The idea of product design activity which results in a useful artifact from diverse parts may be called the design of a system. Whether the system is a machine or a warship, the product of a committee is a complex artifact constructed of diverse parts which are related to each other in some meaningful manner. The product of a committee can be a system in the same sense that the product of a machine shop can be a system. The machine shop is a complex system in the sense that a system is a complex system. The product of a machine shop can be a complex system in the same sense that a system is a complex system. The product of a machine shop can be a complex system in the same sense that a system is a complex system.

The design of a system is a complex activity. It is a process of selecting and arranging diverse parts to achieve a desired goal. The parts are selected and arranged to achieve a desired goal. The process is a complex activity. The process is a complex activity. The process is a complex activity.

The design of a system is a complex activity. It is a process of selecting and arranging diverse parts to achieve a desired goal. The parts are selected and arranged to achieve a desired goal. The process is a complex activity. The process is a complex activity. The process is a complex activity.

The design of a system is a complex activity. It is a process of selecting and arranging diverse parts to achieve a desired goal. The parts are selected and arranged to achieve a desired goal. The process is a complex activity. The process is a complex activity. The process is a complex activity.
Any organization that designs a system (defined more broadly here than just information systems) will inevitably produce a design whose structure is a copy of the organization's communication structure.

-- Mel Conway, 1967
HOW DO COMMITTEES INVENT?

by MELVIN E. CONWAY

A system’s design is a copy of the organization’s communication structure.

design organization criteria

A system’s design is a copy of the organization’s communication structure.

-- Mel Conway, 1967
Communication dictates design.

-- Mel Conway, 1967
Conway’s Law
Brooks’ Law

“Adding manpower to a late software project makes it later.”

-- Fred Brooks, 1975
Intercommunication formula

\[ n(n - 1) / 2 \]

-- Fred Brooks, 1975
Intercommunication formula

\[
\begin{align*}
5 \times (5 - 1) / 2 &= 10 \\
15 \times (15 - 1) / 2 &= 105 \\
50 \times (50 - 1) / 2 &= 1,225 \\
150 \times (150 - 1) / 2 &= 11,175 \\
\end{align*}
\]

-- Fred Brooks, 1975
Dunbar’s Number

A measurement of the “cognitive limit to the number of individuals with whom any one person can maintain stable relationships.”

-- Robin Dunbar, 1992
The Social Cortex

As brain size increases, so does group size. Human group size as predicted by Dunbar’s model comes out to about 150.

Average social group size vs. Size of neocortex relative to rest of brain

- Humans
- Monkeys
- Apes
Dunbar’s Number
the max number of relationships a person can maintain
Dunbar Groups

Intimate friends: 5
Trusted friends: 15
Close friends: 35
Casual friends: 150

-- Robin Dunbar, 1992
Intercommunication formula

\[
5 \times (5-1) / 2 = 10 \\
15 \times (15-1) / 2 = 105 \\
50 \times (50-1) / 2 = 1,225 \\
150 \times (150-1) / 2 = 11,175
\]

-- Fred Brooks, 1975
HOW DO COMMITTEES INVENT?

by MELVIN E. CONWAY

Design organization criteria

"Communication dictates design."

— Mel Conway, 1967
Conway’s (first) Law
So... what about other Conway Laws?
Conway’s Second Law
There is never enough time to do something right, but there is always enough time to do it over.

-- Mel Conway, 1967
Trade Offs
Efficiency-Effectiveness Trade Offs (ETTOs)
The ETTO Principle: Efficiency-Thoroughness Trade-Off
Why Things That Go Right Sometimes Go Wrong

Erich Hollnagel
Conway’s Third Law
Homomorphism

There is a homomorphism from the linear graph of a system to the linear graph of its design organization

-- Mel Conway, 1967
homomorphism

/'hɒməˈmɒrfɪzəm/

noun  MATHEMATICS

a transformation of one set into another that preserves in the second set the relations between elements of the first.
Figure 3 Two examples of identity of structure between a system and its design organization.
Homomorphism

“If you have four groups working on a compiler, you'll get a 4-pass compiler.”

- Eric S. Raymond, 1991
Conway’s Fourth Law
The structures of large systems tend to disintegrate during development, qualitatively more so than with small systems.”

-- Mel Conway, 1967
Three reasons Disintegration occurs...
Disintegration: Reason #1

“The realization that the system will be large, together with organization pressures, make irresistible the temptation to assign too many people to a design effort.”

-- Mel Conway, 1967
Brooks’ Law

Adding manpower to a late software project makes it later.

-- Fred Brooks, 1975
Disintegration: Reason #2

"Application of the conventional wisdom of management to a large design organization causes its communication structure to disintegrate."

-- Mel Conway, 1967
Dunbar’s Number

A measurement of the “cognitive limit to the number of individuals with whom any one person can maintain stable relationships.”

-- Robin Dunbar, 1992
Disintegration: Reason #3

“Homomorphism insures that the structure of the system will reflect the disintegration which has occurred in the design organization.”

-- Mel Conway, 1967
Communication dictates design.

-- Mel Conway, 1967
The machine is the organization.
Scaling
“Free” as in “Scale-Free”
“A scale-free network is a network whose degree distribution follows a power law.”
“A scale-free network is a network whose degree distribution follows a power law.”
$P(k) \sim k^{\Gamma}$
(a) Random network
(a) Random network

(b) Scale-free network
Emergence of Scaling in Random Networks

Albert-László Barabási* and Réka Albert

Department of Physics, University of Notre-Dame, Notre-Dame, IN 46556

Systems as diverse as genetic networks or the world wide web are best described as networks with complex topology. A common property of many large networks is that the vertex connectivities follow a scale-free power-law distribution. This feature is found to be a consequence of the two generic mechanisms that networks expand continuously by the addition of new vertices, and new vertices attach preferentially to already well connected vertices.
This applies to code, too…
Code as hierarchy
The more code, the more bugs.
“The industry average is about 15 - 50 errors per 1000 lines of delivered code.”

- Steve McConnell
Code is not the solution, code is the problem.
the
thing is next the big
thing
is
big
next
the
small
MONOLITHIC/LAYERED

MICRO SERVICES
NO
NO CODE
Artisinal Luddites
Ned Ludd, 1811

The **Luddites** were 19th-century English textile workers who protested against newly developed labor-economizing technologies.

- Wikipedia
Ned Ludd, 1811

The Luddites were 19th-century English textile workers who protested against newly developed labor-economizing technologies.

- Wikipedia
artisanal
/ärˈtēzən(ə)l/

adjective
relating to or characteristic of an artisan.
"artisinal skills"
artisanal
/ərˈtɪzən(ə)l/  adjective
relating to or characteristic of an artisan.
"artisanal skills"
- (of a product, especially food or drink) made in a traditional or non-mechanized way.
"artisanal cheeses"
The slow programming movement is a software development philosophy that emphasizes careful design, quality code, software testing and thinking.

- Wikipedia
Slow Programming

The slow programming movement is a software development philosophy that emphasizes careful design, quality code, software testing and thinking.

- Wikipedia
“What works good is better than what looks good, because what looks good can change, but works good will still work.”

– Charles Eames
“What works good is better than what looks good, because what looks good can change, but works good will still work.”

– Charles Eames
“What works good is better than what looks good, because what looks good can change, but works good will still work.”

– Charles Eames
Exhibition designed for IBM by the Office of Charles and Ray Eames
A COMPUTER PERSPECTIVE
A sequence of 20th century ideas, events, and artifacts from the history of the information machine

Charles & Ray Eames
“I think everybody in this country should learn how to program a computer because it teaches you how to think.”

- Steve Jobs
Life skills are not the same as livelihood.
"People may outlaw driving cars because it's too dangerous. You can't have a person driving a two-ton death machine."

- Elon Musk
ONE NIGHT ONLY
SATURDAY AT THE ARENA:
THRILL to the CONTEST of
MAN VS. MACHINE!

DAVID REES
ARTISANAL PENCILSMITH
"SHARP STEADY HAND"
"THE ONLY STEEL...IS IN HIS NERVE"

PANASONIC KP-310
ELECTRONIC AUTO-STOP
"IT NEVER SLEEPS"
"NO PENCIL IS SAFE FROM ITS JAWS"

WHO WILL BE APPOINTED KING?

CERTIFICATE OF SHARPENING

SAFETY GOGGLES A MUST!!
The Future of Code
The opposite of Artisinal Programming is…
How Etsy Deploys More Than 50 Times a Day

by João Miranda on Mar 17, 2014 | 3 Discuss

Daniel Schauenberg described at the last QCon London how Etsy, renowned for its DevOps and Continuous Delivery practices, does 50 deploys/day. A fully automated deployment pipeline, thorough application monitoring and IRC-based collaboration are all important to achieve this rate of change while keeping risk to a minimum.

Etsy's development approach revolves around making many small, continuous changes. A direct consequence is the need to do many deployments a day. In the words of Daniel Schauenberg, at
How Netflix Deploys Code

by Zef Hemel on Jun 13, 2013 | 3 Discuss

Netflix, the popular movie streaming site, deploys a hundred times per day, without a quality assurance department and without release engineers. Netflix built an advanced in-house PaaS (Platform as a Service) that allows each team to deploy their own part of the infrastructure whenever they want, however many times they require. During QCon New York 2013, Jeremy Edberg gave a talk about the infrastructure built to support this rapid pace of iteration on top of Amazon's AWS.
Part of my job at HubSpot is to meet and welcome new potential customers to our offices. One of the most surprising things I get to tell them is that we deploy our code approximately 300 times per day.

Netflix, the popular movie streaming site, deploys a hundred times per day, without the use of Chef or Puppet, without a quality assurance department and without release engineers. Netflix built an advanced in-house PaaS (Platform as a Service) that allows each team to run their own part of the infrastructure whenever they want, however many times they require. During QCon New York 2013, Jeremy Edberg gave a talk about the infrastructure Netflix built to support this rapid pace of iteration on top of Amazon’s AWS.
What’s going on here?
Software Development at the Speed of DevOps

DevOps seeks to improve the speed and quality of software development and support. It involves integrating and automating the work of software developers and IT operations professionals.

In recent years, some IT organizations have moved to a new software development delivery model called “DevOps.” DevOps refers to a set of practices designed to facilitate interactions between the application development and operations functions, and ultimately aims to increase the speed, improve the quality, and reduce the cost of software development, ongoing enhancement, and support.

Traditionally, IT organizations have run their development and operations functions separately, largely because the two groups have different missions that yield opposing behaviors. “App Dev”
Yep. DevOps.
Yep. DevOps.

But for code.
The First Way:
Systems Thinking

(Business) -> (Customer)

Dev -> Ops

The Three Ways: The Principles Underpinning DevOps
By Gene Kim
The First Way: Systems Thinking

(Business)  

The Second Way: Amplify Feedback Loops

(Customer)  

DevOps

The Three Ways: The Principles Underpinning DevOps
By Gene Kim
The Three Ways: The Principles Underpinning DevOps
By Gene Kim
The Three Ways: The Principles Underpinning DevOps

By Gene Kim
Four Pillars of OPS
Four Pillars of OPS

• Reduce Cost
• Increase Speed
• Improve Safety/Resiliency
• Provide Visibility/Feedback
Reduce Cost

How Netflix Works

- Consumer Electronics
- AWS Cloud Services
- CDN Edge Locations

Customer Device (PC, PS3, TV...)

- Web Site or Discovery API
- User Data
  - Personalization
- Streaming API
  - DRM
  - QoS Logging
- OpenConnect CDN Boxes
  - CDN Management and Steering
  - Content Encoding

@adrianco
Reduce Cost - Virtualize Hardware

How Netflix Works

- Consumer Electronics
- AWS Cloud Services
- CDN Edge Locations

Customer Device (PC, PS3, TV...)

- Web Site or Discovery API
- Streaming API
- OpenConnect CDN Boxes

User Data
- Personalization
- DRM
- QoS Logging
- CDN Management and Steering
- Content Encoding

@adrianc0
Increase Speed
Increase Speed – Automate Deployment

Team → Version control → Continuous integration → Quality gates → Approval → Production
Improve Resiliency

Diagram:
- Automation Testing Process
  - Test Automation Feasibility Analysis
  - Appropriate Tool Selection
  - Develop Test Script, Execute and Analyze
  - Develop Automation Framework
  - Evaluate the suitable framework
  - Build the Proof of Concept
Improve Resiliency – Automated Testing
Provide Visibility
Provide Visibility – Ubiquitous Logging
Same Pillars for DEV
Four Pillars of DEV

- Reduce Cost
- Increase Speed
- Improve Safety/Resiliency
- Provide Visibility/Feedback
Reduce Cost

CA Service Virtualization
Rapidly Create High-Quality Software

SOLUTION

- Quickly create, edit and deploy virtual services
- Patented “Learning Mode” which automatically learns differences between your test and live environments
- Uses artificial intelligence to translate any protocol when creating virtual services

VALUE

- Dramatically reduce the time it takes to build critical business applications
- Avoid delays caused by constrained or unavailable resources
- Enable more comprehensive testing that begins earlier in the cycle so more defects are found sooner to improve quality and reduce costs

SHIFT LEFT

Reduce duration, effort and cost and improve quality by detecting defects here...

Reduce in cost to resolve

COST TO RESOLVE DEFECT

- Unit: $418
- Performance: $779
- UAT: $2,379
- Production: $5,000
Reduce Cost - Virtualize System

CA Service Virtualization
Rapidly Create High-Quality Software

SOLUTION
- Quickly create, edit and deploy virtual services
- Patented “Learning Mode” which automatically learns differences between your test and live environments
- Uses artificial intelligence to translate any protocol when creating virtual services

VALUE
- Dramatically reduce the time it takes to build critical business applications
- Avoid delays caused by constrained or unavailable resources
- Enable more comprehensive testing that begins earlier in the cycle so more defects are found sooner to improve quality and reduce costs

SHIFT LEFT
Reduce duration, effort and cost and improve quality by detecting defects here... instead of here

Reduce in cost to resolve

COST TO RESOLVE DEFECT

<table>
<thead>
<tr>
<th>Stage</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>$418</td>
</tr>
<tr>
<td>Performance</td>
<td>$779</td>
</tr>
<tr>
<td>UAT</td>
<td>$2,379</td>
</tr>
<tr>
<td>Production</td>
<td>$5,000</td>
</tr>
</tbody>
</table>
Increase Speed

```javascript
var claim_code = [];

// This is where the Issuer Gadget is hosted
var baseUrl = "https://sites.google.com/site/<<YOUR SITE>>/";

// The claim code holds the row number and the type pf badge, which for now is
var claim_code_base = "row=" + lastrow;
claim_code.push(Utilitys.base64Encode(claim_code_base + "&type=badge"));

// Build the URL to send
var url = baseUrl + "?claim_code=" + claim_code;

// Compose text for the email
var emailText = "Hi " + name + "!
Congratulations on obtaining the " + badge + "!

// Using the MailApp function of Apps Script to send the email to the person
MailApp.sendEmail(email, "Claim your Badge - " + badge + ", emailText);
```
Increase Speed – Automate Code

```javascript
var claim_code = [];  //This is where the Issuer Gadget is hosted

var baseUrl = "https://sites.google.com/site/<YOUR SITE>/";

// The claim code holds the row number and the type pf badge, which for now is
var claim_code_base = "row=" + lastrow;

claim_code.push(Utility.base64Encode(claim_code_base + "&type=openid"));

// Build the URL to send
var url = baseUrl + "?claim_code=" + claim_code;

// Compose text for the email
var emailText = "Hi " + name + "Congratulation on obtaining the " + badgeName + " badge.";

// Using the MailApp function of Apps Script to send the email to the person
MailApp.sendEmail(email, "Claim your Badge - " + badgeName + ", emailText;
```
Improve Resiliency

source-code

Regular expressions
Finite automata

Lexical Analyzer

Token Stream

Syntax Analyzer

Context-free Grammar

Token Stream
Improve Resiliency – Code Analytics

Diagram:
- Source code
- Lexical Analyzer
- Syntax Analyzer
- Token Stream
- Regular expressions
- Finite automata
- Context-free Grammar
Provide Visibility
Provide Visibility – Runtime Visualization
Summary
Jennings, Wescoff, & Lichterman, 1946
The program was the machine.
The machine was the code.
HOW DO COMMITTEES INVENT?

by MELVIN E. CONWAY

"One kind of intellectual activity which results in a useful whole from its diverse parts may be called the design of a system. Whether the particular activity is the creation of specifications for a major weapons system, the formulation of a communication network, or the preparation of a corporate strategy, the general activity is largely the same.

The traditional object of a design organization is the creation and presentation of a document: a diagram, a map, a chart, a set of specifications, or a story diagram. The system changes it logically produces for a purpose that achieve some larger purpose, usually guided by the system design. The purpose, a useful whole effect, will be proposed and negotiated to meet a requirement of a document design. As the approach to explain the system takes a simpler form and design is made a part of the presentation activity to specify what should be invented.

Our design organization may or may not be involved in the contracting of the design of design. Essentially, it is an activity that is not a result of the system design will be to satisfy the body of work, probably among the design decisions which the system design is required to make. Most design activities are essentially arbitrary. One can also be involved in design decisions which the system design is required to make.

The system design will be to satisfy the body of work, probably among the design decisions which the system design is required to make. Most design activities are essentially arbitrary. One can also be involved in design decisions which the system design is required to make.

Dr. Conner is in management or applied systems research at Honeywell's Information Sciences, where he is working on research in the field of computer science. He has previously been a research associate at the Center for Advanced Study of Behavioral Sciences. When not in research, he has an M.A. in social psychology from Harvard University.
Brooks’ Law

“Adding manpower to a late software project makes it later.”

-- Fred Brooks, 1975
Dunbar Groups

Intimate friends: 5
Trusted friends: 15
Close friends: 35
Casual friends: 150

-- Robin Dunbar, 1992
The machine was the organization.
$P(k) \sim k^\Gamma$
The machine is the network.
Four Pillars of DEV

• Reduce Cost  
  *(Virtualize the System)*

• Increase Speed  
  *(Automate/Eliminate Code)*

• Improve Safety/Resiliency  
  *(Code Analysis)*

• Provide Visibility/Feedback  
  *(Runtime Visualization)*
Bridging the Gap
working smarter in networks

Artisans - Hierarchies - Networks

~19th C  +/- 20th C  21st C
“We must break away from the sequential. We must state definitions and provide for priorities and descriptions of data. **We must state relationships, not procedures.**”

- Grace Hopper, 1962
“Recognizing the need is the primary condition for design.”

– Charles Eames
Those who ignore the mistakes of the future are bound to make them.”
– Dr. Joseph Miller